Fuzzing dans la sphère VoIP

Humberto Abdelnur, Olivier Festor, Radu State

« Fuzzing » dans la sphère VoIP

Humberto Abdelnur, Olivier Festor, Radu State

INRIA Nancy, Grand Est

Résumé

La voix sur IP (VoIP) s'impose aujourd'hui comme l'une des technologies clefs de l'Internet actuel et futur. Dans cet article, nous partageons l'expérience pratique acquise ces deux dernières années par notre équipe de recherche sur l'automatisation des processus de découverte de vulnérabilités dans le monde VoIP. Nous dressons un portrait relativement sombre de la sécurité actuelle de la sphère VoIP en présentant les vulnérabilités les plus dangereuses capables d'aboutir à la compromission de réseaux entiers. Toutes les vulnérabilités présentées dans cet article ont été publiées par notre équipe de recherche et ont été découvertes à l'aide de notre propre suite logicielle de fuzzing appelée KIF. Toute vulnérabilité présentée dans l'article est également accompagnée d'une présentation d'une solution permettant de s'en prémunir.

1/ Introduction

Le fuzzing de protocole s'est imposé ces dernières années comme une approche clef pour la découverte de vulnérabilités dans des implémentations logicielles et matérielles. Le concept sous-jacent au fuzzing est extrêmement simple : générer des données aléatoires et/ou malveillantes et les injecter dans l'application cible par ses divers canaux de communication et d'interaction. Cette approche est différente de la discipline bien établie de tests logiciels. Cette dernière se focalise essentiellement sur les aspects fonctionnels de l'application visée. En fuzzing, le test fonctionnel est marginal, l'objectif d'aboutir rapidement à la découverte de vulnérabilités étant prédominant. Le fuzzing de protocoles est important pour deux raisons majeures. Tout d'abord, disposer d'une approche automatique de découverte de vulnérabilités facilite le processus global d'analyse du programme. Ce processus d'analyse est très souvent complexe et très gourmand en temps. Il nécessite des connaissances profondes en débogage de logiciels ainsi qu'en reverse engineering. De plus, il existe de nombreuses situations dans lesquelles l'accès au code source et/ou au binaire de l'application est impossible. Dans ce cas, seule une approche de test de type « boîte noire » est possible. Le fuzzing protocolaire est applicable à une très grande variété de cibles, allant d'implémentations spécifiques à un équipement propriétaire [9] à des couches présentation [15].

Equipement	**Firmware**
Asterisk	v1.2.16, v1.4.1
	Asterisk-addons v1.2.8 Asterisk-addons v1.4.4
Cisco 7940/7960	vP0S3-07-4-00
	vP0S3-08-6-00

	vP0S3-08-7-00
Cisco CallManager	v5.1.1
FreePBX	v2.3.00
Grandstream Bugde Tone-200	v1.1.1.14
Grandstream GXV-3000	v1.0.1.7
Linksys SPA941	v5.1.5
	v5.1.8
Nokia N95	v12.0.0.13
OpenSer	v1.2.2
Thomson ST2030	v1.52.1
Trixbox	v2.3.1

Table 1 : équipements de l'environnement de test

Nous avons appliqué nos algorithmes de fuzzing intelligent sur de multiples équipements et piles protocolaires SIP utilisés dans des infrastructures Voix sur IP dans le but de tester leur résistance à des situations, des comportements et des données inattendues. Tous les tests présentés ont été effectués à l'aide de la suite logicielle que nous avons développée, décrite dans [8]. Notre approche est basée sur un fuzzing protocolaire à états permettant de traiter des protocoles complexes tels SIP. A notre connaissance, notre environnement est le premier fuzzer SIP capable de dépasser la seule génération de données aléatoires. Notre méthode est basée sur un algorithme d'apprentissage exploitant des traces réseau réelles pour élaborer son automate d'attaque. Cet automate est lui-même en évolution permanente durant le processus de fuzzing. Notre travail est, dans ce domaine, motivé par deux facteurs : premièrement valider par la pratique les contributions formelles dans le monde du fuzzing ; deuxièmement découvrir des vulnérabilités et, en suivant une politique éthique[1] de révélation de celles-ci, aider les équipementiers à corriger les bugs de leurs systèmes et les encourager à notifier les clients potentiellement vulnérables afin qu'ils mettent à jour leurs systèmes.

Nous allons aborder en profondeur ces points dans cet article qui suit le plan suivant : la section 2 présente l'infrastructure VoIP que nous avons utilisée tout au long de notre étude. Les sections suivantes couvrent les différents types de vulnérabilités découvertes, allant de la simple omission de validation de paramètres d'entrée à des failles exploitant de façon combinée plusieurs couches protocolaires et plusieurs technologies. La dernière section conclut l'article et donne des pistes pour des évolutions futures de ce travail.

[1] L'annonce des vulnérabilités est éthique dans le sens où toute vulnérabilité à fait l'objet d'une notification spécifique auprès du constructeur au moins trois mois avant son annonce publique, celle-ci étant dans le cas idéal, faite de façon conjointe entre le constructeur et l'équipe avec fourniture du patch permettant de corriger la vulnérabilité.

2/ « Fuzzing » d'équipements Voix sur IP

Les infrastructures Voix sur IP comprennent un ensemble d'équipements dédiés (en général orientés vers une application) utilisant des technologies de l'Internet comme transport sous-jacent. Les usagers exploitent des équipements terminaux souvent simples (ex : des téléphones) interagissant avec différents types de serveurs afin de gérer les comptes, la mobilité, la localisation et bien sûr l'établissement d'appel entre usagers. L'établissement d'appel est réalisé sur la base d'un protocole de signalisation dont SIP [14] est devenu un des principaux standards, soutenu notamment par l'IETF. Un nombre croissant d'équipements VoIP embarquent aujourd'hui une pile protocolaire SIP en charge du traitement des messages de ce même protocole. Ces piles implantent un automate complexe. Dans la grande majorité des cas, l'accès au code source des piles protocolaires est impossible et pour la plupart des « hardphones » VoIP qui ont des plateformes matérielles spécifiques, aucun moyen de débogage ouvert n'existe pour un chercheur en sécurité indépendant. Dans ce contexte, seule une approche de type test de sécurité de type boîte noire est réalisable dans le cadre d'une activité d'audit de sécurité.

Nous avons effectué nos expérimentations de sécurité et de fuzzing sur une large gamme d'équipements hétérogènes. Les équipements utilisés sont énumérés dans la table 1. Toutes les expérimentations ont été réalisées avec notre environnement logiciel KIF [8]. Dans sa version de base, KIF comprend deux composants autonomes : le fuzzer syntaxique et le fuzzer protocolaire. Ces deux composants fournissent une entité de validation de données à états. Les tests générés peuvent être conforme au comportement (et aux données) normatif ; ils peuvent également inonder l'équipement testé avec des données malveillantes en entrée. De telles données malveillantes peuvent être non conformes à la syntaxe (telle que définie dans la spécification normative des unités de données du protocole) ; elles peuvent également être syntaxiquement conforme mais véhiculer des attaques sémantiques et/ou des contenus malveillants (débordement de tampons, débordements d'entiers, chaînes formatées ou débordements de tas).

Le fuzzer syntaxique a pour objectif unique de générer des messages individuels d'attaque. Il s'appuie pour cela sur la grammaire de ces messages exprimée à l'aide de la métasyntaxe ABNF (Augmented Backus Naur Form)[10] ainsi que sur un scénario de fuzzing. Ce scénario pilote la génération des règles de production dans la grammaire de la syntaxe. Il peut également dépendre du fuzzer protocolaire afin de générer le message final qui sera envoyé à l'entité cible.

Le fuzzer protocolaire effectue du test passif et actif. Pour cela, deux automates sont requis : l'un qui spécifie la machine à états SIP et l'autre qui spécifie la machine à états de l'activité de test. La première machine est utilisée dans le test passif. Elle contrôle l'occurrence de comportement anormal issu de la cible durant la phase de test. Cet automate peut être inféré d'un ensemble de traces SIP relatives à la cible collectées durant des phases opérationnelles normales. Le second automate est utilisé pour du test actif ; il pilote le profil du test de sécurité. Cet automate est défini par l'utilisateur et peut évoluer dans le temps.

L'environnement KIF est illustré dans la figure 1.

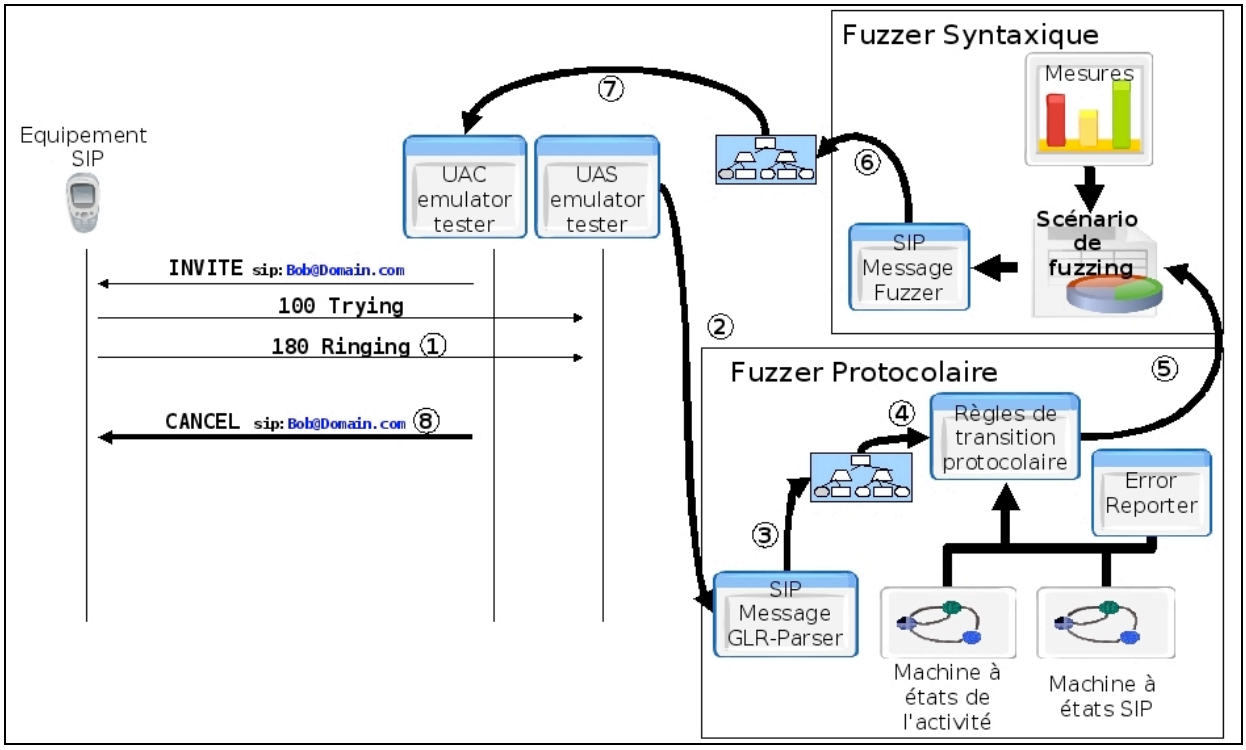

Figure 1 : Environnement KIF

3/ Faiblesses dans la validation des paramètres d'entrée

La vulnérabilité que nous avons rencontrée le plus fréquemment est liée à un filtrage extrêmement faible (voir inexistant) des données fournies en entrée d'entités voix sur IP via le canal SIP. Ce filtrage, lorsqu'il existe, ne traite pas proprement les méta caractères, les caractères spéciaux, les données de grande longueur ou les caractères spécifiques de formatage. Les failles qui en résultent sont dues à des débordements de tampon/tas ou des vulnérabilités de type *« format string ».* La cause la plus probable à cela est que les développeurs de ces systèmes sont partis d'un modèle de menaces dans lequel la signalisation SIP est supposée générée par des piles protocolaires saines et éprouvées. Une raison plus simple encore peut être l'absence dans les processus de conception de certains de ces équipements de toute ou partie de la dimension sécurité. Le véritable danger de cette vulnérabilité provient du fait que dans la grande majorité des cas, un très faible nombre de paquets peut littéralement paralyser un réseau VoIP complet. Ceci est d'autant plus dangereux que dans le cas présent, les messages SIP sont transportés sur UDP, ouvrant la porte à des attaques efficaces effectuées de façon furtive par des techniques simples de spoofing IP. La table 2 illustre un sous-ensemble des vulnérabilités que nous avons publiées ; nous mettons en exergue deux cas extrêmes : la première vulnérabilité (publiée dans CVE-2007-4753) révèle que dans le cas étudié, même le test le plus simple de vérification de l'existence de données en entrée n'est pas effectué. Cette absence de vérification permet des attaques extrêmement simples et efficaces telles que l'envoi d'un paquet vide. Le second cas (CVE-2007-1561) est situé à l'extrême du premier sur l'échelle de la complexité. Ici un serveur VoIP est vulnérable à une attaque dont la structure de données d'entrée est relativement complexe. Le danger repose dans ce cas sur le fait qu'un unique paquet va détruire le serveur voix sur IP de cœur et ainsi rendre indisponible l'ensemble du service VoIP associé. Se prémunir de

telles attaques à un niveau de défense réseau est possible via des techniques d'inspection profonde de paquets couplées à des équipements de filtrage de paquets spécifiques au domaine.

Equipement	Synopsis de la vulnérabilité	Identifiant CVE	Impact
Asterisk v1.4.1	Adresse IP invalide dans le champ SDP	CVE-2007-1561	Déni de service
Cisco 7940/7960 vP0S3-07-4-00	Champ Remote-Party-ID erroné	CVE-2007-1590	Déni de service
Grandstream Bugde Tone-200 v1.1.1.14	Champ WWW-Authenticate erroné	CVE-2007-1590	Déni de service
Linksys SPA941 v5.1.5	Injection du code \377 erroné	CVE-2007-2270	Attaque de type XSS
Thomson ST2030 v1.52.1	Injection de version invalide SIP dans le champ VIA	CVE-2007-4553	Déni de service
	Injection de valeur URI invalide dans le champ TO	CVE-2007-4753	Déni de service
	Paquet vide		Déni de service
Linksys SPA941 v5.1.8	Valeur URI éronnée dans le champ user info	CVE-2007-5411	Attaque de type XSS
Asterisk v1.4.3	Injection de valeur URI invalide dans le champ TO	CVE-2007-54 88	Injection SQL
FreePBX v2.3.00 Trixbox v2.3.1	Injection de valeur URI invalide dans le champ TO	[7]	Attaque de type XSS

Table 2 : Vulnérabilités de validation de paramètres d'entrée

La majorité des équipements voix sur IP embarquent un serveur web, typiquement utilisé pour la configuration, permettre aux utilisateurs de consulter différents journaux (ex : journaux des appels passés, appels manqués, …). Le point important est ici que l'utilisateur peut être amené à consulter les journaux depuis son poste de travail, opérant généralement sur le réseau interne de l'entreprise. Si l'information fournie par les serveurs web des équipements n'est pas efficacement filtrée, l'utilisateur exposera sa machine située dans le réseau interne de l'entreprise à des malwares extrêmement efficace. A titre d'illustration, nous présentons ci-dessous une vulnérabilité découverte au cours d'une campagne de fuzzing (voir CVE-2007-5411). Le téléphone VoIP Linksys SPA-941 (version 5.1.8 du firmware) comprend un serveur web au travers duquel il est possible d'une part de configurer le téléphone et d'autre part de consulter l'historique des appels. Dans ce service, une vulnérabilité de type Cross-Site-Scripting (XSS [11]) permet à un attaquant d'effectuer des injections XSS sur les navigateurs des usagers car

le champ « FROM » des messages SIP n'est pas filtré de façon rigoureuse. En envoyant un message SIP modifié avec le champ « FROM » positionné avec la valeur suivante par exemple :

"<script x='" <sip:'src='http://baloo/beef/y.js'>@192.168.1.9:5060>;tag=1"

Le navigateur de l'usager est redirigé pour inclure un fichier javascript (y.js) depuis une machine externe (baloo) tel que montré dans la figure 2. Cette machine externe est sous le contrôle de l'attaquant et le code javascript injecté, lui permet d'utiliser la machine de l'usager pour scanner le réseau interne de l'entreprise, lancer des attaques de type CSRF (Cross Site Request Forgery), obtenir des informations sensibles (historique des appels, configuration du réseau interne, …) désactiver le pare-feu ou rediriger le navigateur de l'usager vers des pages web infestées de malware (comme par exemple MPACK [2]) dans le but d'infester la machine de la victime. La vulnérabilité provient dans ce cas du fait que l'association de deux technologies (SIP et WEB) est possible sans que la sécurité des flux d'informations entre ces deux technologies n'ait été traitée.

Figure 2 : Attaque XSS contre un équipement Linksys SPA-941

Equipement	Synopsis de la vulnérabilité	Identifiant CVE	Impact
Cisco 7940/7960 vP0S3-08-6-00	Traitement incorrect des messages de type OPTIONS dans une transaction INVITE.	CVE-2007-4459	Déni de service
Grandstream GXV-3000 v1.0.1.7	Traitement incorrect d'un message de type 183 dans une transaction INVITE	CVE-2007-4498	Mise sous écoute
CallManager v5.1.1 OpenSer v1.2.2	Le mécanisme d'authentification permet la ré-utilisation d'un jeton	CVE-2007-5468	Fraude
SIP Protocol Relay Attack	L'attaquent peut se faire passer pour le serveur d'authentification	[6]	Fraude
Cisco 7940/7960 vP0S3-	Traitement incorrect	CVE-2007-5583	Déni de service

08-7-00	d'une chaine de 6 transactions INVITE simultanées.		
Nokia N95 v12.0.013	Traitement incorrect d'un message de type CANCEL inattendu	CVE-2007-6371	Déni de service

Table 3 : Vulnérabilités dans des états protocolaires avancés

L'impact de cette vulnérabilité est très fort : la plupart des pare-feux et systèmes de prévention d'intrusion ne protègent pas le réseau interne d'attaques XSS menées au travers de SIP. De plus, les usagers se connectent à ces équipements directement depuis le réseau interne de l'entreprise et de ce fait le rendent vulnérable. Jeremiah Grossmann [11] a montré comment les pare-feux peuvent être désactivés avec des attaques XSS. De nombreuses autres attaques malveillantes existent ici. Malheureusement, la plupart des équipements VoIP embarquent des serveurs et applications Web faibles de telle sorte que d'autres systèmes vulnérables existent et sont sans aucun doute exploités.

4/ Vulnérabilités de suivi protocolaire

Les vulnérabilités de suivi protocolaire vont au delà du simple filtrage d'un unique message SIP. Dans ce type de vulnérabilités, plusieurs messages vont amener un équipement cible dans un état inconsistant ; tout message utilisé dans cette chaîne d'attaque considéré en isolation ne violera pas la spécification normative du protocole SIP [14]. Ces vulnérabilités proviennent en grande majorité d'une faiblesse dans l'implémentation des automates du protocole. Elles peuvent être exploitées de trois façons différentes :

1. L'équipement peut recevoir des entrées qui ne sont pas attendues dans l'état courant du protocole : par exemple en envoyant au système un BYE alors qu'il s'attend à recevoir un INVITE,
2. L'entrée peut prendre la forme de messages simultanés dirigés vers plusieurs états du protocole,
3. De faibles variations dans les champs de suivi de dialogues et/ou transaction SIP peuvent amener un équipement vers un état inconsistant.

La découverte de telles vulnérabilités est un problème difficile. Le processus de fuzzing doit ici être capable d'identifier où et à quel moment un équipement cible ne suit pas rigoureusement le protocole et quels champs des messages peuvent être « fuzzés » pour révéler la vulnérabilité. L'espace de recherche est dans ce cas gigantesque, couvrant de multiples messages et champs de données ; l'utilisation de techniques de fuzzing avancées pilotées par des méthodes d'apprentissage est ici indispensable. Le tableau 3 comprend une liste des vulnérabilités que nous avons publiées dans cette famille. Comme pour le cas précédent (vulnérabilités liées au filtrage des données), les vulnérabilités présentées sont de complexité variable.

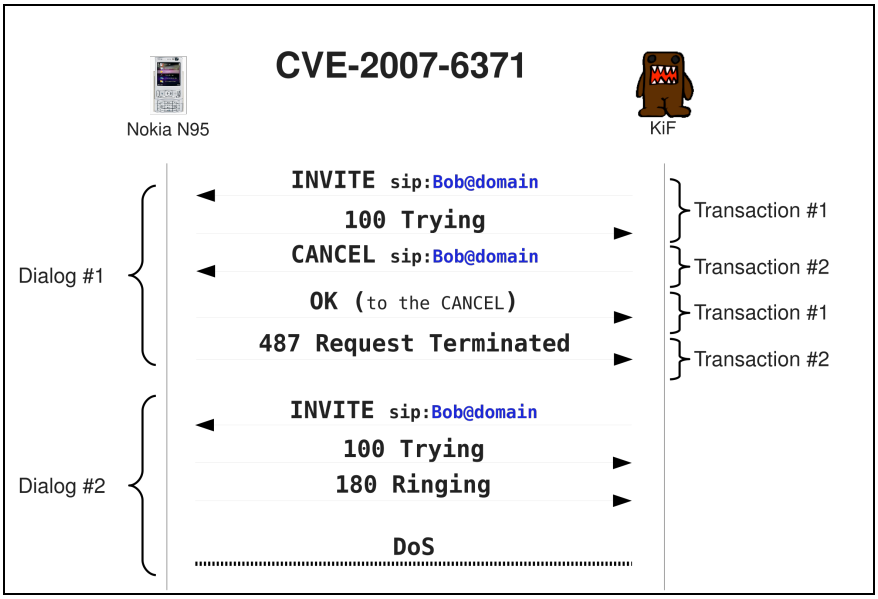

Figure 3 : Attaque de type déni de service sur un téléphone Nokia N95

Un cas simple est celui de la (CVE-2007-6371). Ici, l'envoi prématuré d'un message CANCEL peut amener l'équipement dans un état inconsistant qui aboutit à un déni de service comme illustré dans la figure 3. Le danger majeur de ce type d'attaques est qu'à ce jour, aucun pare-feu applicatif ne peut suivre et inspecter un si grand nombre de flux et que même dans le cas où les signatures sont connues, des versions polymorphiques d'attaques efficaces peuvent aisément être obtenues et ainsi passer entre les mailles des systèmes de protection. A ce jour malheureusement, aucune solution efficace pour la prévention de ce type d'attaque n'existe.

4.1 Fraude à la facturation

Une fraude à la facturation intervient lorsque la véritable source d'un appel (ou la destination lorsqu'il s'agit d'un numéro vert par exemple) n'est pas facturée. Ceci peut se produire en utilisant une infrastructure VoIP compromise ou en manipulant le trafic de signalisation. Il est très surprenant de constater que malgré l'évolution sans précédent des technologies, les astuces basiques élaborées dans les années 1970, où les phreakers reproduisaient le signal à 2600 Hz utilisé par les opérateurs, continuent à fonctionner. Près de 40 ans après, le plan de signalisation peut toujours être manipulé et attaqué par des utilisateurs mal intentionnés. Ce qui a changé cependant, est la technologie pour le faire. De nos jours, il est possible d'injecter des commandes SQL (Chapitre VI [13]) dans le plan de signalisation et la fraude devient possible. Dans la suite de cette section, nous détaillons une vulnérabilité de ce type découverte lors d'une campagne de fuzzing dans notre laboratoire [7]. Certains proxys SIP stockent de l'information collectée des entêtes SIP dans des bases de données. Ceci est nécessaire pour les fonctions de comptabilité et de facturation. Si cette information n'est pas proprement filtrée, elle pourra lors de sa visualisation par l'administrateur, effectuer une injection SQL de second ordre, i.e. les données visualisées sont interprétées comme un code SQL par l'application et celui-ci est exécuté en tant que tel. Il en résulte la possibilité de modifier la base de données à l'aide du code injecté. Dans ce cas (modification de la base de données), un attaquant

peut par exemple facilement réduire la durée enregistrée de tous les appels afin de minimiser la facture des appelants. Si l'on considère le populaire et largement déployé PbX VoIP Asterisk, les enregistrements d'appel (CDR : Call Detail Records) sont stockés dans une base MySQL. FreePBX [1] et Tribox [5] utilisent l'information disponible dans cette base de données pour gérer et générer les factures et/ou la charge du PBX.

Plusieurs fonctions ne traitent pas correctement les échappements pour tous les caractères dans les champs des messages de signalisation. Une première déclinaison de cette attaque spécifique peut être déclenchée par un utilisateur reconnu du domaine cible. Celui-ci n'a qu'à injecter des nombres négatifs dans les tables d'enregistrement des détails d'appels afin de changer la durée ou tout autre paramètre d'un appel donné. La conséquence directe de cela est qu'en raison de la maîtrise par l'attaquant des données exploitées par les services de comptabilité et de facturation, ces processus tombent eux-mêmes sous le contrôle total de l'attaquant. Une seconde conséquence encore plus sérieuse vient du fait que cette attaque peut être étendue en injectant des balises JavaScript [11] afin qu'elles soient exécutées par le poste de l'administrateur lorsqu'il/elle effectue des opérations de maintenance de base. De cette situation résulte la possibilité d'une attaque de type Cross-Site Scripting (XSS) car du code Javascript malveillant aura pu être stocké dans la base de données via l'injection SQL. Ce malware sera exécuté dans le navigateur de l'administrateur lorsque celui-ci accèdera aux enregistrements. Ce processus est similaire aux attaques d'injection dans des logs, bien connues dans la communauté sécurité des applications Web. De façon similaire au cas précédent, des outils tels que Beef et XSS proxy peuvent scanner le réseau interne, désactiver les pare-feux et déclencher toutes les attaques CSRF/XSRF spécifiques.

Le problème principal est que la plupart des applications qui manipulent des enregistrements CDR ne considèrent pas ce type d'attaque. De plus, si le système cible n'est pas parfaitement sécurisé, les injections SQL peuvent aboutir à la compromission de l'ensemble du système cible car la plupart des serveurs de bases de données autorisent des interactions avec le système d'exploitation cible [13].

Ce type de vulnérabilité est dangereux car comme nous l'avons déjà indiqué ci-dessus, peu d'applications (en fait aucune de celles que nous avons testées à ce jour) implémentent du filtrage au niveau des entêtes SIP. Toutes les applications considèrent que les informations qui proviennent de messages SIP sont issues d'une source fiable ou à défaut, bienveillante ! Se prémunir de ce type d'attaque requiert un filtrage efficace des données tant en entrée qu'en sortie à chaque fois qu'une information est lue/enregistrée depuis/vers un autre composant logiciel.

4.2 Ecoutes distantes

Durant une campagne de fuzzing, nous avons découvert une vulnérabilité aussi surprenante qu'inattendue. Elle est révélée dans (CVE-2007-4498). Ici, plusieurs messages SIP envoyés au terminal cible, activent le microphone du téléphone, ouvrent les canaux voix depuis la cible vers l'attaquant sans bien sûr qu'aucun élément ne permette à l'usager de voir que son téléphone est décroché ni qu'une communication est en cours ; le rêve pour une écoute distante ! En effet, l'attaquant peut ainsi suivre toutes les communications vocales dans le site de la victime. L'échange protocolaire qui réalise

cette attaque est décrit dans la figure 4. L'impact de cette vulnérabilité dépasse la « simple » écoute d'appels voix sur IP car elle permet de placer à coût nul ou presque un micro dans une salle sans effraction et de suivre l'intégralité des conversations qui s'y déroulent. Le risque est majeur et devrait être pris en compte lors de toute décision du choix du fournisseur et du déploiement d'un équipement Voix sur IP. Bien que dans le cas présent, la vulnérabilité résulte probablement d'une erreur de programmation, de telles portes laissées consciemment ouvertes par des entités ou équipementiers mal intentionnés représentent de véritables menaces.

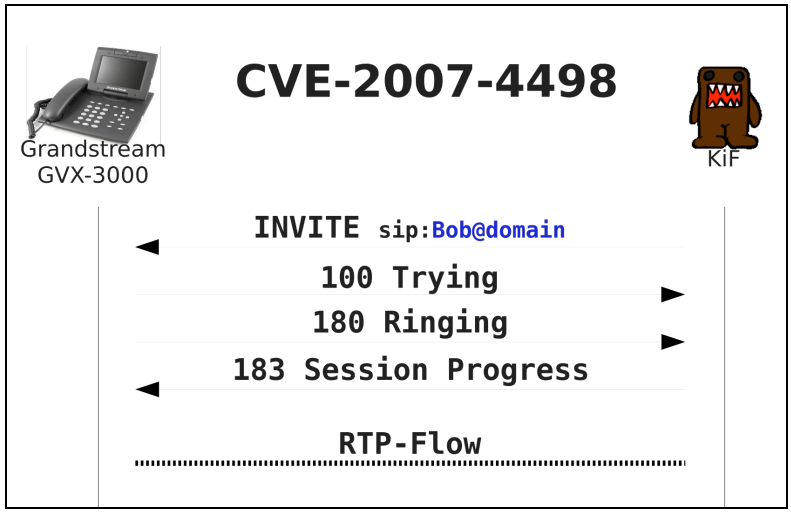

Figure 4 : Ecoute distante via un téléphone Grandstream GXV-3000

4.3 Implémentation cryptographiques faibles

Le mécanisme d'authentification de SIP est basé sur un secret partagé et un challenge/réponse [12]. Des nonces sont générés par le serveur et soumis à l'entité souhaitant s'authentifier. Celle-ci doit utiliser la clef partagée pour calculer un hash, lui-même envoyé au serveur. Ce hash est calculé sur plusieurs valeurs : les entêtes SIP et des nonces. Un hash reçu par le serveur est validé par celui-ci et utilisé pour authentifier le client. Pour des raisons d'efficacité, peu de serveurs suivent le cycle-de-vie (notamment la durée de validité) d'un jeton. Nous avons découvert deux vulnérabilités (CVE-2007-5468 et CVE-2007-5469) dans lesquelles des jetons interceptés peuvent être réutilisés. Ces vulnérabilités ne sont pas de simples attaques de type man-in-the-middle car les jetons sont ici réutilisables durant de longues périodes et ont pu être utilisées pour l'authentification et l'établissement de multiples autres appels que ceux pour lesquels ils avaient été générés. La figure 5 présente le flux des messages pour de telles attaques. Comme pour les précédentes, l'impact de telles attaques et fort. Les escroqueries à la facturation et le vol d'identifiants d'appels en sont des conséquences immédiates. Se prémunir de telles attaques nécessite de revoir le compromis entre performance et sécurité et requiert l'implémentation de procédures de gestion des jetons cryptographiques sûres et performantes.

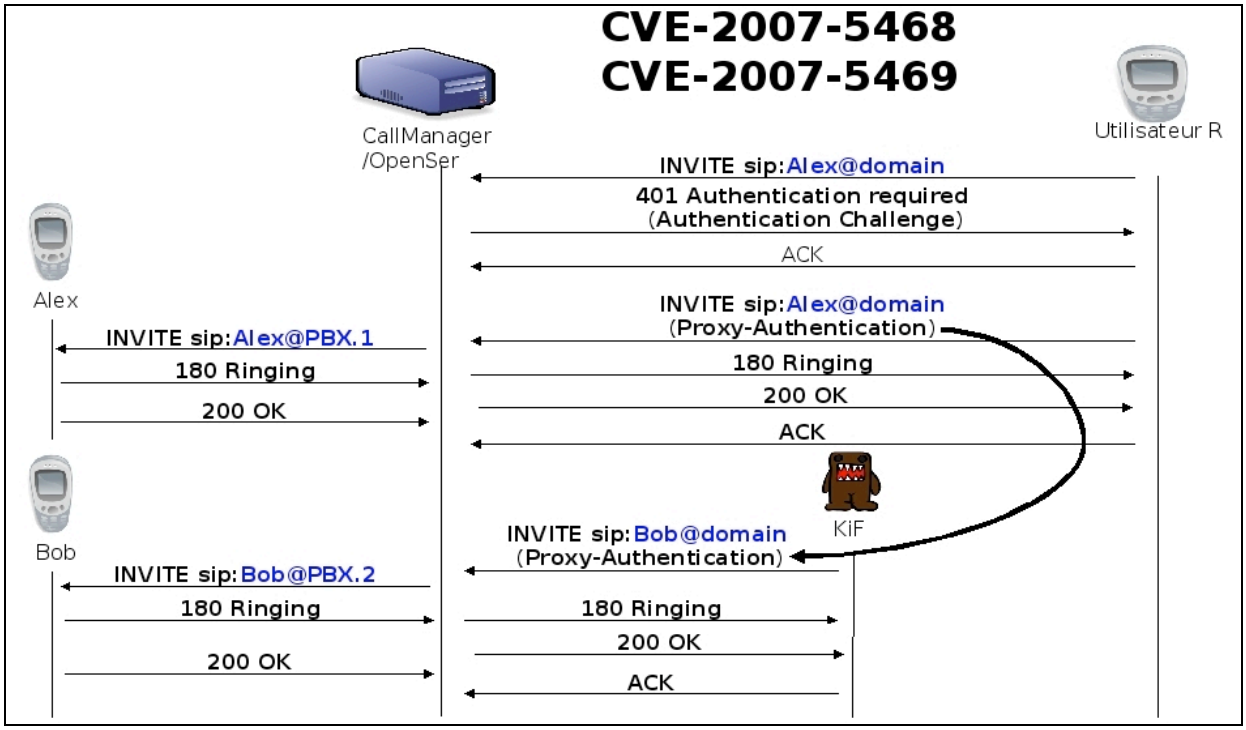

Figure 5 : Attaque de rejeu étendue : dans cette configuration, l'attaquant (KIF) récupère les données d'authentification transmises par un utilisateur légal (utilisateur R) lors d'un appel à Alex. Le vil attaquant réutilise ces données afin d'appeler un tiers (ici Bob) en se faisant passer pour l'utilisateur légal.

5/ Vulnérabilités dans les spécifications du protocole

Nous avons consacré une part importante de notre activité à la recherche de vulnérabilités sur des implémentations spécifiques du protocole SIP sans initialement considérer la sécurité du protocole en soi. C'est lors de l'exécution d'un scénario de fuzzing complexe qui nous avons relevé la même anomalie (et vulnérabilité apparente) sur tous les équipements sous test (tous ceux répertoriés dans la table 1). Ceci nous à naturellement conduit à lancer une analyse sur la spécification du protocole SIP, notamment en utilisant des techniques formelles et outils supports tels AVISPA[2]. Cette analyse nous a permis d'identifier la vulnérabilité dans la conception même du protocole, vulnérabilité qui rend toute attaque d'escroquerie à la facturation possible sur tout réseau voix sur IP [6]. Le problème vient en effet du fait qu'une attaque classique de type relais est possible en forçant une entité appelée à émettre un message de type RE-INVITE. Cette attaque étant nouvelle, générique et sévère, elle est naturellement dangereuse.

Voici comment elle se matérialise : un attaquant établit un appel avec sa victime. Sa victime répond (décroche) et est amenée à mettre l'appelant en attente (il existe plusieurs méthodes pour la conduire à entreprendre cette action, la plus simple étant qu'un complice appelle la victime alors que celle-ci est en communication avec l'attaquant). Lorsque l'attaquant reçoit le message SIP re-invite qui spécifie la mise en attente, celui-ci peut demander à la victime de s'authentifier. Cette dernière

[2] http://avispa-project.org/

authentification peut être utilisée par l'attaquant pour se substituer à la victime sur son propre proxy. Détaillons maintenant l'attaque en utilisant la notation suivante :

- P est le proxy localisé à l'URL : proxy.org
- X est l'attaquant localisé à l'URL : attaquant.lan.org
- V est la victime localisée à : victime.lan.org
- V est également enregistré auprès de P sous le nom d'utilisateur victime@proxy.org
- Y est le complice de X. Physiquement cela peut être X lui-même mais nous utilisons une autre notation pour des raisons de clarté.

Nous montrons comment X appelle le numéro surtaxé 0-800-xx-xx-xx à l'insu de V qui sera facturé.

1. X appelle directement V.

   Le champ RECORD-ROUTE doit comporter l'adresse de l'attaquant et le champ contact doit avoir comme valeur l'identifiant du numéro surtaxé que l'attaquant veut appeler à l'insu de la victime. Ceci permet de préparer l'attaque en positionnant chez la victime les champs clef : destination des échanges de signalisation (ici l'attaquant par le champ record-route) et numéro surtaxé (ici placé dans le champ contact). Ceci donne la requête suivante :

```
X ---------- INVITE victime.lan.org -------------> V
     From : attaquant à attaquant.lan.org
     To: victime à victime.lan.org
     Contact: 1900-XXXX à proxy.org
     Record-Route: attaquant.lan.org
```

2. Le déroulement normal des opérations SIP

```
X <--------------- 180 Ringing ------------------ V
X <----------------- 200 OK --------------------- V
X <--------------- Media Data ------------------> V
```

3. Le complice Y entre en jeu et invite la victime V. Celle-ci décide de mettre X en attente ;
4. La victime V envoie un message RE-INVITE à X (pour lui indiquer sa mise en attente tel que défini dans la section 12.2.1.1 du RFC de SIP [14]) ;

```
X <----------- INVITE 190XXXX at proxy.org ------ V
     From: victime à victime.lan.org
     To : attaquant à attaquant.lan.org
```

5. X appelle le numéro surtaxé en utilisant le proxy P qui lui demande de s'authentifier en utilisant un digest d'authentification avec comme nonce "Proxy-Nonce-T1" et comme realm= "proxy.org" ;
6. X demande à V d'authentifier le message RE-INVITE qu'il a reçu de la victime à l'étape 4 et utilise dans sa demande les mêmes paramètres de digest et de realm que ceux demandés par le serveur à X durant l'étape 5;

```
X ------------401/407 Authenticate -------------> V
```

```
Digest: realm ="proxy.org",
        nonce="Proxy-Nonce-T1"
```

7. A cette étape l'attaque est bouclée ; la victime va faire pour X (et lui transmettre) le travail d'authentification auprès du proxy (attaque relai) ;

```
X <---------- INVITE 190XXXX at proxy.org -------- V
     Digest: realm ="proxy.org", nonce="Proxy-Nonce-T1"
             username= "victime",
             uri="1900XXXX at proxy.org",
             response="the victime computed response"
```

8. X peut désormais répondre au proxy avec un digest parfaitement valide (construit par V) et qui authentifie V.

6/ Conclusions et travaux futurs

Les résultats quantitatifs et qualitatifs issus des campagnes de recherche de vulnérabilités que nous avons menées sont éloquents. Absolument tous les équipements que nous avons testés sont vulnérables et le spectre des vulnérabilités trouvées est très large. Des vulnérabilités de validation triviale de paramètres sont courantes et elles affectent des équipements particulièrement sensibles. Des vulnérabilités dans des états protocolaires avancés existent également, elles sont seulement beaucoup plus complexes à mettre à jour. La cause principale de toutes ces vulnérabilités repose essentiellement sur une faible prise en compte de la sécurité dans le cycle de vie de leurs équipements. L'intégration des technologies du Web et de la voix sur IP est une véritable boîte de Pandore qui renferme des dangers bien plus complexes et puissants. Des attaques spécifiques au Web peuvent être initiées et menées à bout au travers du plan de signalisation SIP pouvant aboutir à des effets dévastateurs, tel que la prise de contrôle totale d'un réseau interne de l'entreprise ou de celui de son fournisseur de services. Ceci est possible car aucun pare-feu applicatif ne permet à ce jour d'interagir avec de multiples technologies afin de fournir des gardes assurant l'interaction sûre entre les mondes du Web et de la voix sur IP. Une cause plus structurelle à cet état est l'absence d'un modèle de menaces réel et complet pour la voix sur IP. L'alliance VOIPSA a développé un modèle de menaces [4] qui ne reflète cependant pas l'état réel des menaces actuelles. Nous avons montré au travers des vulnérabilités découvertes que des attaques de déni de service extrêmement dévastatrices peuvent être réalisées avec un nombre limité de paquets sans éveiller le moindre soupçon dans les systèmes de prévention et/ou de détection d'attaques ; l'espionnage à distance dépasse le cadre minimal de l'interception d'une simple communication et le plan de signalisation SIP couplé aux services construits au dessus apparaît lui-même comme un vecteur de menaces pour l'ensemble de l'infrastructure de communication et du système d'information.

Il reste beaucoup à faire pour assainir la situation. Le principal objectif étant d'aboutir à des équipements voix sur IP intégrant une sécurité forte. Des modifications dans le cycle de développement de ces équipements doivent impérativement intégrer des phases d'audit et de test de sécurité. Le fuzzing de

protocoles est un élément essentiel de ces phases d'autant plus qu'il représente un moyen unique pour des chercheurs en sécurité indépendants d'évaluer les équipements. Dans les échanges que nous avons eu avec les différents constructeurs, un seul a réagi de façon extrêmement professionnelle en intégrant systématiquement les vulnérabilités remontées dans son processus internet d'évolution de ses produits et en travaillant avec nous sur la publication conjointe des vulnérabilités et des patchs de correctifs. Les communautés du logiciel libre en Voix sur IP ont sur ce point également été très réactives à nous découvertes et ont corrigé les vulnérabilités dans des temps records. Pour la majorité des constructeurs dont nous avons testé des équipements, la prise en compte des vulnérabilités découvertes dans le processus d'évolution des systèmes reste encore souvent tabou et/ou inexistante.

Nous avons dans cet article décrit une partie de l'expérience pratique acquise par le groupe sur le test de piles protocolaires SIP au travers de campagnes de fuzzing intense. Nous avons réalisé ces tests dans le seul but de valider nos travaux de recherche sur de nouveaux algorithmes et modèles de fuzzing à des fins de sécurité. Les résultats ont largement dépassé nos espérances et nous encouragent à poursuivre nos investigations dans ce sens. Toutes les vulnérabilités présentées ici ont été révélées dans des conditions claires et respectueuses des règles établies dans la communauté de sécurité. Nous poursuivons nos travaux sur deux points : l'un porte sur les systèmes de protection contre les attaques, le second sur les modèles et algorithmes de découverte de vulnérabilités. Sur le premier point nous avons notamment développé un pare-feu SIP supportant le suivi d'échanges complexes permettant d'anticiper des attaques. Ce pare-feu répondant au doux nom de SECSIP est distribué sous License GPL2 au travers de la Gforge de l'INRIA. Dans le domaine des méthodes et algorithmes de fuzzing, nous travaillons actuellement à la généralisation des méthodes utilisées pour SIP et à leur application à d'autres protocoles.

Références bibliographiques

[1] FreePBX: full-featured PBX web application. http://freepbx.org

[2] MPack: Insight into MPACK Hacker kit . http://www.malwarehelp.org/news/article- 6268.html

[3] The Asterisk PBX. http://www.asterisk.org/

[4] The Voice over IP Security Alliance (VOIPSA). http://www.voipsa.org/Activities/taxonomy.php

[5] trixbox: Asterisk-based IP-PBX . http://www.trixbox.com/

[6] H. Abdelnur R. State O. Festor. Security Advisory: "SIP Digest Access Authentication RELAY-ATTACK for Toll-Fraud''. http://voipsa.org/pipermail/voipsec_voipsa.org/2007- November/002475.html

[7] H. Abdelnur R. State O. Festor. Security Advisory: "SQL injection in asterisk-addons

and XSS injection in WWW application in Areski, FreePBX and Trixbox''. http://voipsa.org/pipermail/voipsec_voipsa.org/2007- October/002466.html

[8] Humberto Abdelnur Radu State Olivier Festor. "KiF: A stateful SIP Fuzzer". In Proceedings of Principles, Systems and Applications of IP Telecommunications, IPTComm, pages 47–56, New-York, NY, USA, JUL 2007. ACM Press.

[9] Laurent Butti and Julien Tinnes. Discovering and exploiting 802.11 wireless vulnerabilities. Journal in Computer Virology, 4(1):25–37, February 2008.

[10] D. Crocker and P. Overell editors, "Augmented BNF for Syntax Specifications: ABNF". RFC 5234, STD 68, January 2008.

[11] Seth Fogie, Jeremiah Grossman, Robert Hansen, Anton Rager, and Petko D. Petkov. XSS Exploits: Cross Site Scripting Attacks and Defense. Syngress, 2007.

[12] Alan B. Johnston and David M. Piscitello. Understanding Voice over Ip Security. Artech, 2006.

[13] David Litchfield, Chris Anley, John Heasman, and Bill Grindlay. The Database Hacker's Handbook: Defending Database Servers. John Wiley & Sons, 2005.

[14] H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. "SIP: Session Initiation Protocol". http://www.ietf.org/rfc/rfc3261.txt, June 2002.

[15] Michael Sutton, Adam Greene, and Pedram Amini. Fuzzing: Brute Force Vulnerability Discovery. Addison-Wesley Professional, 2007.